

AMENDMENTS TO THE CLAIMS

Please amend claims 1, 14, 18, 21-24, and 27-30 as follows, without acquiescence or prejudice to pursue the original claims in a related application. No new matter has been added. A complete listing of the current pending claims is provided below and supersedes all previous claims listing(s).

1. (Currently Amended) A method for simultaneous debugging of an electrical design having both an HDL portion and a general programming language portion, comprising:

interrupting a simulator ~~handling that operates upon~~ the HDL portion, the simulator interrupted by an external debugger, the external debugger for debugging the general programming language portion;

handling a simulator request with ~~an~~ the external debugger for the simulator that is interrupted, the external debugger calling a request processing function at the simulator, the simulator request for simulation of the HDL; ~~and~~

executing the request processing function at the simulator to respond to the simulator request; and

generating debug results based upon executing the request processing function and storing the debug results in a computer-readable medium.

2. (Original) The method of claim 1 in which the simulator request accesses a portion of the HDL portion.

3. (Original) The method of claim 2 in which the simulator request accesses HDL signal values.

4. (Original) The method of claim 2 in which the simulator request accesses HDL design hierarchy.

5. (Original) The method of claim 1 in which the simulator request operates simulator functionality.

6. (Original) The method of claim 1 in which the general programming language portion comprises C, C++, or SystemC code.

7. (Original) The method of claim 1 in which the HDL portion comprises VHDL or Verilog.
8. (Original) The method of claim 1 in which the action of having the external debugger call the request processing function is based upon recognition of a waiting simulator request.
9. (Original) The method of claim 8 in which recognition of the waiting simulator request is based upon a message sent to the external debugger.
10. (Original) The method of claim 8 in which recognition of the waiting simulator request is based upon a periodic check of a simulator request wait queue.
11. (Original) The method of claim 8 in which recognition of the waiting simulator request is based on whether a threshold number of simulator requests are waiting in a simulator request wait queue.
12. (Original) The method of claim 1 in which the simulator request is generated at a simulator GUI.
13. (Original) The method of claim 12 in which the response to the simulator request is displayed at the simulator GUI.
14. (Currently Amended) The method of claim 1, wherein ~~in which~~ the external debugger ~~that~~ calls the request processing function at the simulator is the gdb debugger, ~~with the following statement:~~

 ~~call expr~~
15. (Original) The method of claim 1 in which the simulator request is routed through a debugger GUI for the external debugger.
16. (Original) The method of claim 1 in which the simulator request is directly routed to the external debugger.
17. (Original) The method of claim 1 in which the request processing function is set up ahead of time at the simulator to handle anticipated simulator requests.
18. (Currently Amended) A method for simultaneous processing of a design that is based upon multiple programming languages, the multiple programming languages comprising a first

language portion and a second language portion, in which processing of the second language portion interrupts processing of the first language portion, the method comprising:

processing the second language portion of the design causing an interruption of processing for the first language portion;

determining whether there are one or more waiting requests for processing of the first language portion;

handling the one or more waiting requests for processing of the first language portion by having processing of the second language portion call a request processing function at the first language portion that has been interrupted; and

executing the request processing function at the first language portion to process the one or more waiting requests; and

generating processing results based upon executing the request processing function and storing the processing results in a computer-readable medium..

19. (Original) The method of claim 18 in which the one or more waiting requests are for accessing data from the first language portion of the design.

20. (Original) The method of claim 18 in which the one or more waiting requests are for debugging the first language portion.

21. (Currently Amended) The method of claim 18, wherein the act of determining whether there are one or more waiting requests for processing of the first language portion is based upon a message sent to a debugger for the processing of the second language portion.

22. (Currently Amended) The method of claim 18, wherein the act of determining whether there are one or more waiting requests for processing of the first language portion is based a periodic check of a request wait queue for the first language portion.

23. (Currently Amended) The method of claim 18, wherein the act of determining whether there are one or more waiting requests for processing of the first language portion is based on whether a threshold number of simulator requests are waiting in a request wait queue.

24. (Currently Amended) The method of claim 18, ~~wherein in which~~ the request processing function is called by a gdb debugger, ~~with the following statement:~~

~~call expr~~

25. (Original) The method of claim 18 in which processing the second language portion comprises debugging the second language portion.

26. (Original) The method of claim 18 in which the request processing function is set up ahead of time to handle anticipated requests.

27. (Currently Amended) A computer program product comprising a volatile or non-volatile computer usable medium having executable code to execute a process for simultaneous debugging of an electrical design having both an HDL portion and a general programming language portion, the process comprising:

interrupting a simulator ~~handling that operates upon~~ the HDL portion, the simulator interrupted by an external debugger, the external debugger ~~for debugging the general~~ programming language portion;

handling a simulator request with ~~an~~ the external debugger for the simulator that is interrupted, the external debugger calling a request processing function at the simulator, the simulator request for simulation of the HDL; and

executing the request processing function at the simulator to respond to the simulator request; and

generating debug results based upon executing the request processing function and storing the debug results in a computer-readable medium.

28. (Currently Amended) A system for simultaneous debugging of an electrical design having both an HDL portion and a general programming language portion, comprising:

means for interrupting a simulator ~~handling that operates upon~~ the HDL portion, the simulator interrupted by an external debugger, the external debugger ~~for debugging the general~~ programming language portion;

means for handling a simulator request with an the external debugger for the simulator that is interrupted, the external debugger calling a request processing function at the simulator, the simulator request for simulation of the HDL; and

means for executing the request processing function at the simulator to respond to the simulator request; and

means for generating debug results based upon executing the request processing function and storing the debug results in a computer-readable medium.

29. (Currently Amended) A computer program product comprising a volatile or non-volatile computer usable medium having executable code to execute a method for simultaneous processing of a design that is based upon multiple programming languages, the multiple programming languages comprising a first language portion and a second language portion, in which processing of the second language portion interrupts processing of the first language portion, the method comprising:

processing the second language portion of the design causing an interruption of processing for the first language portion;

determining whether there are one or more waiting requests for processing of the first language portion;

handling the one or more waiting requests for processing of the first language portion by having processing of the second language portion call a request processing function at the first language portion that has been interrupted; and

executing the request processing function at the first language portion to process the one or more waiting requests; and

generating processing results based upon executing the request processing function and storing the processing results in a computer-readable medium.

30. (Currently Amended) A system for simultaneous processing of a design that is based upon multiple programming languages, the multiple programming languages comprising a first language portion and a second language portion, in which processing of the second language portion interrupts processing of the first language portion, the ~~method~~ system comprising:

means for processing the second language portion of the design causing an interruption of processing for the first language portion;

means for determining whether there are one or more waiting requests for processing of the first language portion;

means for handling the one or more waiting requests for processing of the first language portion by having processing of the second language portion call a request processing function at the first language portion that has been interrupted; and

means for executing the request processing function at the first language portion to process the one or more waiting requests.